UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/924,260	08/08/2001	Evelyn Duesterwald	10011525-1	4900

7590 09/21/2005

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
FOWLKES, ANDRE R	
ART UNIT	**PAPER NUMBER**
2192	

DATE MAILED: 09/21/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/924,260	DUESTERWALD ET AL.	
	Examiner	Art Unit	
	Andre R. Fowlkes	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 July 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20 and 23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20 & 23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 7/12/05 has been entered.

Claim Objections

2. The objection to claim 22 is withdrawn, in view of applicant's amendment.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-20 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bugnion, U.S. Patent No. 6,704,925 in view of Yates et al., (Yates), U.S. Patent No. 5,802,373.

As per claim 1, Bugnion discloses an **apparatus for dynamically transforming and caching at least one computer program,** (col. 1:61-65, "Dynamic binary translators perform the translation from an original instruction sequence to a host instruction sequence during the execution of the program. The translated code sequences are then stored in a buffer called the translation cache"), **the apparatus comprising:**

- one or more computer readable storage media (col. 4:27-28, "The output instruction sequences are stored in a translation cache (computer readable storage media)"),

- computer executable instructions stored in the one or more computer readable storage media, (col. 4:27-28, "The output instruction sequences are stored in a translation cache"), **the computer executable instructions comprising:**

- instructions for dynamically transforming code fragments from a plurality of software emulators, (col. 1:61-65, "Dynamic binary translators perform the translation from an original instruction sequence to a host instruction sequence during the execution of the program. The translated code sequences are then stored in a buffer called the translation cache"),

- instructions for caching said code fragments (col. 4:27-28, "The output instruction sequences are stored in a translation cache"),

- instructions providing an application programming interface enabling said at least one computer program to activate said instructions for dynamically transforming said code fragments and said instructions for caching said code

fragments (col. 6:3, Bugnion discloses the applications necessary to communicate with the "operating system" or some other system or control program programs (i.e. application programming interfaces)).

Bugnion doesn't explicitly disclose **instructions for causing said code fragments to be executed simultaneously by at least one computer processor, wherein said plurality of software emulators emulate computer systems with at least two different instruction set architectures.**

However, Yates, in an analogous environment, discloses **instructions for causing said code fragments to be executed simultaneously by at least one computer processor, wherein said plurality of software emulators emulate computer systems with at least two different instruction set architectures** (col. 15:45-48, "emulate multiple addressing spaces (simultaneously executing code fragments) which are possible in the Intel architecture and other non-native architectures").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Yates into the system of Bugnion to have **instructions for causing said code fragments to be executed simultaneously by at least one computer processor, wherein said plurality of software emulators emulate computer systems with at least two different instruction set architectures.** The modification would have been obvious because

one of ordinary skill in the art would have wanted the flexibility and efficiency gains from simultaneously controlling multiple emulators with different instruction set architectures.

As per claim 2, the rejection of claim 1 is incorporated and further, Bugnion discloses that **said instructions providing an application programming interface enable said at least one computer program to provide said code fragments for said instructions for dynamically transforming code fragments and for said instructions for caching said code fragments** (col. 6:3, Bugnion discloses applications to communicate with the "operating system" or some other system or control program programs (i.e. application programming interfaces), and col. 1:61-65, "Dynamic binary translators perform the translation from an original instruction sequence to a host instruction sequence during the execution of the program. The translated code sequences are then stored in a buffer called the translation cache").

As per claim 3, the rejection of claim 1 is incorporated and further, Bugnion discloses that **said instructions providing an application programming interface include providing functions for caching and executing a specified code fragment** (col. 6:3, Bugnion discloses applications to communicate with the "operating system" or some other system or control program programs (i.e. application programming interfaces), and col. 1:61-65, "Dynamic binary translators perform the translation from an original instruction sequence to a host instruction sequence during the execution of

the program. The translated code sequences are then stored in a buffer called the translation cache").

As per claim 4, the rejection of claim 1 is incorporated and further, Bugnion discloses that **said instructions providing an application programming interface include providing functions for configuring behavior of said instructions for dynamically transforming said code fragments and said instructions for caching said code fragments** (col. 6:3, Bugnion discloses an application to communicate with the "operating system" or some other system or control program programs (i.e. application programming interfaces), and functions for configuring the translation system)

As per claim 5, the rejection of claim 1 is incorporated and further, Bugnion discloses that **said instructions for dynamically transforming said code fragments comprise instructions for changing memory address references in said code fragments** (col. 3:38-40, "translate each data reference from a virtual address issued by the simulated processor to a physical address").

As per claim 6, the rejection of claim 1 is incorporated and further, Bugnion discloses that **said instructions for dynamically transforming said code fragments comprise instructions for changing the layout of said code fragments while preserving the function of said code fragments** (col. 8:1, "compiler optimizations

(e.g. code motion, which changes the layout of code, while preserving the original function").

As per claim 7, the rejection of claim 1 is incorporated and further, Bugnion discloses that **said instructions providing an application programming interface include instructions for accessing code fragments a cross a network** (col. 6:3, "computer system (i.e. networked computers", and col. 6:3, Bugnion discloses an application to communicate with the "operating system" or some other system or control program programs (i.e. application programming interfaces)).

As per claim 8, the rejection of claim 1 is incorporated and further, Bugnion discloses that **at least one computer program comprises at least one emulator** (col. 4:25-26, "(a computer programs that) emulate the corresponding input instruction sequences").

As per claim 9, the rejection of claim 1 is incorporated and further, Bugnion doesn't explicitly disclose that **at least one computer program comprises a plurality of emulators being executed simultaneously.**

However, Yates, in an analogous environment, discloses that **at least one computer program comprises a plurality of emulators being executed simultaneously** (col. 15:45-48, "emulate multiple addressing spaces which are possible in the Intel architecture and other non-native architectures").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Yates into the system of Bugnion to have **at least one computer program comprises a plurality of emulators being executed simultaneously.** The modification would have been obvious because one of ordinary skill in the art would have wanted the option of executing multiple emulators for different computer architectures at the same time.

As per claim 10, the rejection of claim 9 is incorporated and further, Bugnion discloses that **said plurality of emulators comprise emulators for at least two different computer architectures** (col. 4:25-26, "(a computer programs that) emulate the corresponding input instruction sequences", and the phrase "Depending on the system architecture", at col. 10:14 , indicates several different computer architectures).

As per claim 11, the rejection of claim 1 is incorporated and further, Bugnion discloses that **at least one computer program comprises at least one operating system** (col. 6:3, "operating system").

As per claim 12, the rejection of claim 1 is incorporated and further, Bugnion doesn't explicitly disclose that said computer executable instructions further comprising instructions for **transparently obtaining said code fragments from said at least one computer program** for said instructions for dynamically transforming said code fragments and for said instructions for caching said code fragments.

However, Yates, in an analogous environment, discloses that said computer executable instructions further comprising instructions for **transparently obtaining said code fragments from said at least one computer program** for said instructions for dynamically transforming said code fragments and for said instructions for caching said code fragments (col. 15:11-12, "to transparently emulate the execution of a ... non-native instruction").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Yates into the system of Bugnion to have a instructions for **transparently obtaining said code fragments from said at least one computer program.** The modification would have been obvious because one of ordinary skill in the art would be motivated to allow existing software to be translated and executed on new hardware while excluding the details from the software user in order to minimize distractions and unnecessary details for a new or novice computer user.

As per claim 13, the rejection of claim 12 is incorporated and further, Bugnion discloses that **said computer executable instructions further comprising instructions for controlling the execution of said at least one computer program on said at least one computer processor** (col. 1:61-65, "Dynamic binary translators perform the translation from an original instruction sequence (i.e. instructions for controlling the execution of a program on a processor) to a host instruction sequence during the execution of the program").

As per claim 14, the rejection of claim 12 is incorporated and further, Bugnion discloses that **said computer executable instructions further comprising instructions for obtaining optimal portions of code from said at least one computer program to create said code fragments** (col. 1:61-65, "Dynamic binary translators perform the translation from an original instruction sequence (i.e. instructions for controlling the execution of a program on a processor) to a host instruction sequence during the execution of the program", and col. 8:1, "(code) optimizations").

As per claim 15, the rejection of claim 1 is incorporated and further, Bugnion discloses that **said instructions for transparently obtaining said code fragments from said at least one computer program obtain said code fragments across a network** (col. 6:3, "computer system (i.e. networked computers", and col. 6:3, Bugnion discloses an application to communicate with the "operating system" or some other system or control program programs (i.e. application programming interfaces)).

As per claim 16, the rejection of claim 1 is incorporated and further, Bugnion discloses that **said computer executable instructions further comprising instructions for optimizing said code fragments** (col. 8:1, "(code) optimizations").

As per claim 17, the rejection of claim 1 is incorporated and further, Bugnion discloses that **said computer executable instructions further comprising**

instructions for linking a plurality of said code fragments to create a larger code fragment (col. 8:1, "(code) optimizations (e.g. replacing a function call, with the actual code for the function, creating a larger, more efficient code fragment").

As per claim 18, the rejection of claim 1 is incorporated and further, Bugnion doesn't explicitly disclose that **said computer executable instructions further comprising instructions for replacing hardware control code in said code fragments, where said hardware control code is adapted to control hardware which is not present and hardware which is not functioning**

However, Yates, in an analogous environment, discloses that **said computer executable instructions further comprising instructions for replacing hardware control code in said code fragments, where said hardware control code is adapted to control hardware which is not present and hardware which is not functioning** (col. 1:15-18, "(the system) includes all of the software resources needed by the computer system to interface each of the hardware elements to the computer system", and these instructions are then translated (i.e. replaced)).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Yates into the system of Bugnion to have **said computer executable instructions further comprising instructions for replacing hardware control code in said code fragments, where said hardware control code is adapted to control hardware which not present and hardware which is not functioning.** The modification would have been obvious

because one of ordinary skill in the art would be motivated to allow the software to exhibit all of the new hardware functionality when it is translated to a different architecture system.

As per claim 19, this is another apparatus version of the claimed apparatus discussed above, in claims 12, 16 and 18 wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Bugnion fig. 2, translation system and Yates col. 15:11-12 and col. 1:15-18.

As per claim 20, this is another apparatus version of the claimed apparatus discussed below, in claim 23, wherein all claimed limitations have also been addressed and/or cited as set forth below. For example, see Bugnion fig. 2, translation system and associated text.

As per claim 23, the rejection of claim 9 is incorporated and further, Bugnion doesn't explicitly disclose that **said plurality of emulators are executed simultaneously on said at least one computer processor using a single instantiation of said instructions for dynamically transforming code fragments.**

However, Yates, in an analogous environment, discloses that **said plurality of emulators are executed simultaneously on said at least one computer processor using a single instantiation of said instructions fro dynamically transforming code fragments** (col. 15:45-48, "emulate multiple addressing spaces (using a single

instantiation) which are possible in the Intel architecture and other non-native architectures").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Yates into the system of Bugnion to have **said plurality of emulators are executed simultaneously on said at least one computer processor using a single instantiation of said instructions fro dynamically transforming code fragments.** The modification would have been obvious because one of ordinary skill in the art would have wanted the convenience of controlling multiple emulators with a single instantiation.

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andre R. Fowlkes whose telephone number is (571) 272-3697. The examiner can normally be reached on Monday - Friday, 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

ARF

TUAN DAM
SUPERVISORY PATENT EXAMINER